最大クリーク問題の多項式時間的可解性について (計算機科学とアルゴリズムの数理的基礎とその応用)

中西, 裕陽; 富田, 悦次; 若月, 光夫; 西野, 哲朗

数理解析研究所講究録
第 1744 巻 2011 年 169-176

最大クリーク問題の多項式時間的可解性について

中西 裕陽* 富田 悦次† 若月 光夫‡ 西野 哲朗§

1 まえがき

対象グラフ中に所定サイズのクリークが存在するか否かを判定する最大クリーク問題は，典型的なNP完全問題であり，多項式時間的に本問題を解くことはほぼ不可能であると強く予測されている [1], [2]．ここで，パーフェクトグラフ [3] 他，幾つかの特殊グラフにおいては最大クリーク問題が多項式時間的に可解であることが明らかにされている（文献 [4] 中の 文献 参照)．ただし，これら特殊グラフに対して最大クリーク問題が実用上高速に解くことが可能とは必ずしも限らない（たとえば，文献 [3] では，8. Conclusion として，Althoug these algorithms are polynomial (and thus are theoretically good) we do not recommend them for practical use. と記述されている)．

これに対し，特殊グラフについてではなく，一般グラフに対して出来る限り自然で緩やかな条件として，どの様な場合ならばこの NP 完全問題が多項式時間的に可解と証明出来るかを示すことは，この他幾多の NP 完全問題との関連においても重要な問題となる．

これに沿った具体的定量結果として，一般グラフにおいて，まず最大次数 Δ が定数である場合に，最大クリーク問題は，節点数 n のグラフに対して $O(n\log n)$ 時間で可解であることが発表されている [5].

また，筆者らは，節点数 n のグラフ中の極大クリークを $O(3^{n/3})$ 時間で全列挙するアルゴリズム CLIQUES[6] を基礎として，一般グラフにおいて，最大次数 Δ が節点数 n の対数オーダである場合に，最大クリーク問題は多項式時間的に可解であることの，次の定量的一基本結果を示した [4].

「節点数 n の一般グラフにおいて，最大次数 Δ が $\Delta \leq 2.495d\lg n$ ($d \geq 1$: 定数) なる条件を満たしている時，このグラフの最大クリーク問題は $O(n^{2+d})$ なる多項式時間で可解である．」

さらに，その定量的改良として，次の結果を与えた [7].

「節点数 n の一般グラフにおいて，最大次数 Δ が $\Delta \leq 2.613d\lg n$ ($d \geq 1$: 定数) なる条件を満たしている時，このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である．」

なお，上記においては，グラフは隣接行列表現 (データ構造) を用いることを前提としていたが ([7]，4章，p.33，右，4行目)，全体的明示が欠けていたので，ここに改めてその前提を追記する．ここで，文献 [4] の基本結果から文献 [7] への改良は，“部分問題の統合による探索領域削減”という新たな手法の導入による探索領域削減の達成によっている．

本稿においては，“部分問題の統合による探索領域削減”をより詳しく解析することにより，更に探索領域の削減を進め，新たに次の結果を与える．

「節点数 n の一般グラフにおいて，最大次数 Δ が $\Delta \leq 2.704d\lg n$ ($d \geq 0$: 定数) なる条件を満たしている時，このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である．」

ここでは，グラフの次数に関する条件の定数部分を 2.613 から 2.704 に増大させると共に，定数 d に対する条件を，$d \geq 1$ から $d \geq 0$ へと拡張している．後者の拡張は，グラフの表現を隣接行列とするとの前提を撤廃する（即ち，隣接リストを用いる）ことにより得ている．

本論文のアルゴリズム，解析手法は，文献 [6], [4], [7] を直接的基盤としており，重複した記述は出来る限り省略している．従って，必要に応じて適宜前出文献を参照いただきたい．

*電気通信大学 先進アルゴリズム研究ステーション
†電気通信大学 先進アルゴリズム研究ステーション；中央大学研究開発機構
‡電気通信大学 先進アルゴリズム研究ステーション；電気通信大学情報理工学研究科
§第 3 著者に同じ

2 諸定義と記法

・本稿で対象とするグラフは，自己閉路をもたない無向グラフ $G=(V,E)$ とする．ここで，V は節点の有限集合，E は相異なる 2 節点の非順序対 (v,w)（これを枝と呼ぶ）の集合である．節点 v と w は，$(v,w)\in E$ が成り立つとき隣接しているという．

集合 V に対して，その要素数を $|V|$ で表す．また，集合 V が順序付き集合である時，その先頭から i 番目の要素を $V[i]$ で表す．

・$v\in V$ について，

$\Gamma(v)=\{w\in V \mid (v,w)\in E\}$ ($\not\ni v$)

と定義し，$|\Gamma(v)|$ を v の次数と呼ぶ．

また，集合 $V=\{v_1,v_2,...,v_n\}$ に対して，

$\hat{\Gamma}(V)=\Gamma(v_1)\cap\Gamma(v_2)\cap...\cap\Gamma(v_n)$

と定義する．

・与えられた $Q\subseteq V$ の誘導部分グラフ $G(Q)$ に対して，次が成り立つとき，$G(Q)$ はクリークであるという．

${}^{\forall}v,w\in Q$ $(v\neq w)$ に対して $(v,w)\in E$.

このとき単に，Q はクリークであるともいう．

節点の部分集合 $W\subseteq V$ による誘導部分グラフ $G(W)$ 中の最大クリークのサイズ (節点数) を $\omega(W)$ で表す．

3 改良アルゴリズム MCP'_1

文献 [4] の基本アルゴリズム MCP_0 を拡張して，文献 [7] のアルゴリズム MCP_1 を得ているが，本稿においてはさらにそれを改良したアルゴリズム MCP'_1 を提唱する．この基本部分を図 1 に示す．ここで，図 1 の 1600 行目 における

$i:=1$ **to** $|EXT_u|-2$

を

$i:=1$ **to** $|EXT_u|-1$ とした時，文献 [4] のアルゴリズム MCP_0(文献 [4]，図 1) とほぼ同じであり，深さ優先探索手続き EXPAND() の適用により，サイズ 0 のクリークから出発して，より大きいクリークを逐次求めてゆく過程が主体である．

改良アルゴリズム MCP'_1 の基本部分全体は，MCP_0 と同様に，下記の限定操作

・部分森の同一化

・探索領域の単純削減 (文献 [4] においては，3.3 EXT_u 最後尾節点の探索省略)

を加えた，深さ優先探索の分枝限定アルゴリズムとしている．

部分森の同一化は，与えられた入力節点集合 $SUBG$ に対して，その最大次数節点 u を先頭として先に探索を行うことにより，u の隣接節点集合 $SUBG_u=SUBG\cap\Gamma(u)$ 中の各節点については探索を省略する，Bron-Kerbosch[8] におけるのと同様の限定操作である．

3.1 探索領域の単純削減

MCP'_1 による探索においては，探索候補節点集合 $SUBG$ 中の最大次数節点を u として，

$SUBG_u=SUBG\cap\Gamma(u)$,

および，

$EXT_u=SUBG-\{u\}-SUBG_u$

を定める．更に，この u，および EXT_u 中の各節点

$u,v_1,v_2,...,v_{|EXT_u|}$

に対して，それぞれ隣接部分

$SUBG_u, SUBG_{v_1}, SUBG_{v_2},..., SUBG_{v_{|EXT_u|}}$

が定義される．このとき節点 v_i は

$SUBG-\{u,v_1,v_2,,..,v_{i-1}\}$ の先頭節点であり，

$SUBG_{v_i}=\Gamma(v_i)\cap(SUBG-\{u,v_1,v_2,,..,v_{i-1}\})$

である．これらの隣接部分は上記の順序に従って探索を行なう．

このとき，$EXT_u=\{v_1,v_2,...,v_{|EXT_u|}\}$ の最後尾節点 $v_{|EXT_u|}$ の隣接部分について，次が成立する．

[補題 1] (文献 [4]，[補題 1] 参照．)

$\omega(\{v_{|EXT_u|}\}\cup SUBG_{v_{|EXT_u|}})\leq\omega(\{u\}\cup SUBG_u)$.

□

これより，EXT_u の最後尾節点 $v_{|EXT_u|}$ からの探索は行なわず，探索領域の単純な削減が出来る．

この様な探索領域の削減を，より強力化することを，次において考える．

3.2 部分問題の統合による探索領域削減

以下では，$|EXT_u|\geq 2$ である場合において，EXT_u 中の最後尾から 2 番目の節点 $v_{|EXT_u|-1}$ について考察する．

[補題 2] $(v_{|EXT_u|-1}, v_{|EXT_u|})\notin E$ であるならば，$\omega(\{v_{|EXT_u|-1}\}\cup SUBG_{v_{|EXT_u|-1}})\leq\omega(\{u\}\cup SUBG_u)$.

```
procedure MCP'_1(G)
begin
   global |Q| := 0;
   global |Q max| := 0;
    EXPAND(V)
end {of MCP'_1}
   procedure EXPAND(SUBG)
   begin
    if SUBG ≠ ∅
    then u := a vertex in SUBG
             that maximizes |SUBG ∩ Γ(u)|;
    ROOT := {u};
         |Q| := |Q| ∪ |ROOT|;
         SUBG_u := Γ(u) ∩ SUBG;
         EXT_u := SUBG − {u} − SUBG_u;
         EXPAND(SUBG_u)
         |Q| := |Q| − |ROOT|;
         for i := 1 to |EXT_u| − 2
         do v_i := the first vertex in EXT_u;
         ROOT := {v_i};
         SUBG_{v_i} := Γ(v_i) ∩ (EXT_u ∪ SUBG_u)
         |Q| := |Q| + |ROOT|;
         EXPAND(SUBG_{v_i});
         |Q| := |Q| − |ROOT|;
         EXT_u := EXT_u − {v_i};
         od
    else {i.e., SUBG = ∅}
      if |Q| ≥ |Q max|
      then |Q max| := |Q|
      fi
   end {of EXPAND}
```

図 1: アルゴリズム MCP'_1 (基本部分)

```
if |EXT_u| ≥ 2
and (EXT_u[|EXT_u| − 1], EXT_u[|EXT_u|]]) ∈ E then
ROOT := {EXT_u[|EXT_u| − 1], EXT_u[|EXT_u|]]};
    SUBG_TAIL :=
      Γ^ ({EXT_u[|EXT_u| − 1], EXT_u[|EXT_u|]})
                              ∩(EXT_u ∪ SUBG_u);
    if |SUBG_TAIL| = |SUBG_u| − 1 then
         SUBG_u := ∅;
    fi
fi
```

```
    |Q| := |Q| + |ROOT|;
    EXPAND(SUBG_TAIL);
    |Q| := |Q| − |ROOT|
```

図 2: アルゴリズム MCP'_1 (探索領域削減)

(証明) 文献 [7], [補題 2] 証明参照. □

この場合には，前節と同様にして，節点 $v_{|EXT_u|-1}$ からの探索も省略して，探索領域の単純削減を行う.

そうでない場合には，この節点からの探索を単純に省略することは出来ない. そこで，その様な場合の状況について考察する.

[補題 3] $\omega(\{v_{|EXT_u|-1}\} \cup SUBG_{v_{|EXT_u|-1}})$
$> \omega(\{u\} \cup SUBG_u)$
なる不等式が成立するのは，$\{v_{|EXT_u|-1}\} \cup SUBG_{v_{|EXT_u|-1}}$ 中の最大クリークが節点 $v_{|EXT_u|}$ を含む場合に限られる.

(証明) 文献 [7], [補題 3] 証明参照. □

補題 3 から，以下に示す操作を導入することにより，$\{v_{|EXT_u|-1}\} \cup SUBG_{v_{|EXT_u|-1}}$ 中に最大クリークが存在する場合, それらを検出することができる.

[1] もし EXT_u の最後尾 2 節点が互いに隣接しないならば, 補題 2 により EXT_u の最後尾 2 節点の探索は単純に省略する.

[2] 補題 3 中の不等式が成立のとき，$\{v_{|EXT_u|-1}\} \cup SUBG_{v_{|EXT_u|-1}}$ 中の最大クリークは節点 $v_{|EXT_u|}$ を含んでいなければならない. そこで最大クリークを抽出するためには, あらかじめ節点 $v_{|EXT_u|}$ を含む集合

$$SUBG_{TAIL} = \hat{\Gamma}(\{EXT_u[|EXT_u|-1], EXT_u[|EXT_u|]\}) \cap (SUBG - \{u, v_1, v_2, ..., v_{|EXT_u|-2}\})$$

についてだけ探索を行えば十分である.

$SUBG_{TAIL}$ は，一般に $SUBG_{v_{|EXT_u|-1}}$ および $SUBG_{v_{|EXT_u|}}$ よりサイズが小さい. 即ち, 次の補題が成り立つ.

[補題 4] 節点数 $|V| = n$ なるグラフ $G = (V, E)$ の最大次数を $\Delta \leq n-1$ とする. 節点の部分集合 $SUBG \subseteq V$ に対して,

$|SUBG_u| = |SUBG \cap \Gamma(u)| = \Delta - k \ (0 \leq k \leq \Delta)$

が最大となる $SUBG$ 中の節点を u として，$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, ..., v_{|EXT_u|-1}, v_{|EXT_u|}\}$ とする. このとき以下が成立する.

$(v_{|EXT_u|-1}, v_{|EXT_u|}) \in E$ ならば,

$|\hat{\Gamma}(\{v_{|EXT_u|-1}, v_{|EXT_u|}\}) \cap SUBG| \leq \Delta - k - 1.$

(証明) 文献 [7], [補題 4] 証明参照. □

補題 4 においては, 集合 $SUBG_{TAIL}$ のサイズの単純な上限を与えた. しかし，$SUBG_{TAIL}$ の内容に関して詳しく考察することにより，さらに探索の削減が実現出来る.

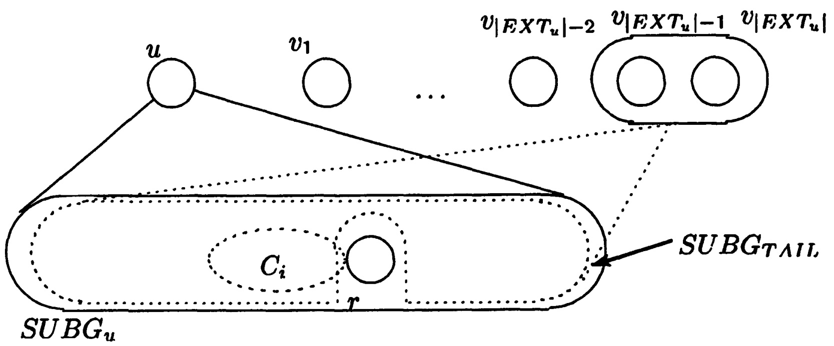

図 3: 補題 5

先ず，以下の関係が成り立つ．

[補題 5] 節点数 $|V| = n$ なるグラフ $G = (V, E)$ の最大次数を $\Delta \le n-1$ とする．節点の部分集合 $SUBG \subseteq V$ に対して，

$|SUBG_u| = |SUBG \cap \Gamma(u)| = \Delta - k$ $(0 \le k \le \Delta)$ が最大となる $SUBG$ 中の節点を u として，$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \ldots, v_{|EXT_u|-1}, v_{|EXT_u|}\}$

$SUBG_{v_i} = \Gamma(v_i) \cap (SUBG - \{u, v_1, v_2, ,.., v_{i-1}\})$ $(1 \le i \le |EXT_u|)$

および

$$SUBG_{TAIL} = \hat{\Gamma}(\{EXT_u[|EXT_u|-1], EXT_u[|EXT_u|]\}) \cap (SUBG - \{u, v_1, v_2, \ldots, v_{|EXT_u|-2}\})$$

とする．このとき以下が成立する．

$(v_{|EXT_u|-1}, v_{|EXT_u|}) \in E$ かつ $|SUBG_{TAIL}| = \Delta - k - 1$ のとき，

$\omega(\{u\} \cup SUBG_u) \le \omega(\{v_{|EXT_u|-1}, v_{|EXT_u|}\} \cup SUBG_{TAIL}))$.

(証明). $SUBG_{TAIL}$

$$= \hat{\Gamma}(\{v_{|EXT_u|-1}, v_{|EXT_u|}\}) \cap (SUBG - \{u, v_1, v_2, \ldots, v_{|EXT_u|-2}\})$$

$$= \hat{\Gamma}(\{v_{|EXT_u|-1}, v_{|EXT_u|}\}) \cap (\{v_{|EXT_u|-1}, v_{|EXT_u|}\} \cup SUBG_u)$$

$$= \hat{\Gamma}(\{v_{|EXT_u|-1}, v_{|EXT_u|}\}) \cap SUBG_u$$

$$\subseteq SUBG_u.$$

また，

$$|SUBG_{TAIL}| = \Delta - k - 1$$

である．したがって，$SUBG_{TAIL} (\subseteq SUBG_u)$ に含まれない $SUBG_u$ 中の節点はただ 1 個存在する．この節点を r とする．

いま $SUBG_u$ 中の最大サイズのクリークを $C_1, C_2, \ldots, C_k$ $(|C_1| = |C_2| = \ldots = |C_k| = \omega(SUBG_u))$ とおく．このとき節点 r は

[1]$C_1, C_2, \ldots, C_k$ のうち少なくとも 1 個に含まれない．

[2]$C_1, C_2, \ldots, C_k$ の全てに含まれる．

のいずれかを満足する．

r が [1] を満たす場合，$C_1, C_2, \ldots, C_k$ のうち r を含まないものの 1 個を C_i $(1 \le i \le k)$ とする．$SUBG_{TAIL}$ は r を除く $SUBG_u$ 中の全ての節点を含むから，

$C_i \subseteq SUBG_{TAIL}$.

従って $\omega(\{v_{|EXT_u|-1}, v_{|EXT_u|}\} \cup SUBG_{TAIL}) = |C_i| + 2 > \omega(\{u\} \cup SUBG_u)$ である．

r が [2] を満たす場合，$SUBG_{TAIL}$ は

$C_1 - \{r\}, C_2 - \{r\}, \ldots, C_k - \{r\}$

なるサイズ $\omega(SUBG_u) - 1$ のクリークを含む．いま $SUBG_{TAIL} (\subseteq SUBG_u)$ は r を除く $SUBG_u$ 中の全ての節点に隣接するのであるから，これらのクリーク中の節点は全て $v_{|EXT_u|-1}$ および $v_{|EXT_u|}$ に隣接する．従って

$\omega(\{v_{|EXT_u|-1}, v_{|EXT_u|}\} \cup SUBG_{TAIL}) = C_i - 1 + 2 = \omega(\{u\} \cup SUBG_u)$

であり，この場合も題意は成立する．□

補題 5 を利用して，アルゴリズムに以下の限定操作を導入する．

(a)$|SUBG_{TAIL}| = |SUBG_u| - 1$ の場合:

常に $SUBG_{TAIL}$ 中に $SUBG_u$ 中の最大クリーク

以上のサイズのものが存在する．そこでこの場合 $SUBG_u$ に対する探索を省略するために，$SUBG_u := \emptyset$ と設定する．

(b)$SUBG_{TAIL} \leq |SUBG_u| - 2$ の場合:

この場合は $SUBG_u$ と $SUBG_{TAIL}$ 両方に探索を行う．

以上の操作全体を合わせて，改めて拡張した意味において“部分問題の統合による探索領域削減”と呼ぶ．

アルゴリズム上でこの処理を実現するために，図2の探索領域削減部分 1301-1308 行を 1300-1400 行間に，2401-2403 行を 2400-2500 行間にそれぞれ挿入する．

以上，図1，2を合わせて得られるアルゴリズム全体を，本稿の改良アルゴリズム MCP'_1 とする．

4 最大時間計算量評価

節点数 n のグラフ $G = (V, E)$ $(|V| = n)$ に対する EXPAND(V) の最大時間計算量を $T(n)$ とする．ここで，任意の $SUBG \subseteq V$ に対する EXPAND($SUBG$) の最大時間計算量 $T(|SUBG|)$ について，次の関係式が成り立つ．

$$\begin{aligned} &T(|SUBG|) \\ &\leq T(|SUBG_u|) + \textstyle\sum_{i=1}^{|EXT_u|-2} T(|SUBG_{v_i}|) \\ &\qquad + T(|SUBG_{TAIL}|) \\ &\qquad\quad + C|SUBG| \cdot (|SUBG_u| + 1). \quad (1) \end{aligned}$$

ここで，u は $SUBG$ において最初に選択する最大次数節点とする．また，

$SUBG_u = \Gamma(u) \cap SUBG$,

$$\begin{aligned} EXT_u &= SUBG - \{u\} - SUBG_u \\ &= \{v_1, v_2, ..., v_{|EXT_u|}\}, \end{aligned}$$

$SUBG_{v_i} = \Gamma(v_i) \cap ((EXT_u - \{v_1, v_2, ..., v_{i-1}\}) \cup SUBG_u)$,

および，

$$\begin{aligned} SUBG_{TAIL} = \hat{\Gamma}\ (\{&EXT_u[|EXT_u| - 1], \\ &EXT_u[|EXT_u|]\}) \end{aligned}$$
$\cap (SUBG - \{u, v_1, v_2, ..., v_{|EXT_u|-2}\})$

とそれぞれおく．

この式末尾の $C|SUBG| \cdot (|SUBG_u| + 1)$ は，このような問題分割に必要な全ての処理に要する多項式オーダの計算量の上界であり，定数 $C > 0$ は以下の様に定義する．ただしここでは，グラフの表現 (データ構造) に隣接リストを採用する．

いま，既に得られている最大クリークを $Q\max \subseteq V$ とすると，探索のある時点において EXPAND() が行う処理の手数は，その時点で極大クリークが得られているかどうかによって変る．そこで，以下の場合分けを行う．

(i)$SUBG \neq \emptyset$ の場合

このとき，EXPAND() はまず最大次数節点 u の選出，集合 $SUBG_u$ と EXT_u の定義，および EXT_u の最後尾 2 節点を用いて集合 $SUBG_{TAIL}$ の定義を行う．

以下，各 $SUBG \subseteq V$ において $SUBG$ 中の節点の隣接関係は，次のように隣接リストを用いて与えるものとする．即ち，各探索深さにおいてサイズ $|SUBG|$ の1次元配列を用意し，これを $SUBG$ の節点を表現する普遍集合とする．この各節点 $v \in SUBG$ の隣接節点集合 $\Gamma(v)$ を抽出して，$SUBG$ 中の各節点の隣接リストを生成する．

このとき各節点に対する隣接節点のリストの長さは高々$|SUBG_u| + 1$ であることから上記処理に要する手数は $O(|SUBG| \cdot (|SUBG_u| + 1))$ である．これより，そのような上界 $C_1|SUBG| \cdot (|SUBG_u| + 1)$ を与える定数を C_1 とする．

$EXT_u = SUBG - \{u\} - SUBG_u$ を生成する操作は $O(|SUBG_u| + 1)$ の手数で可能である．そこで上界 $C_2(|SUBG_u| + 1)$ を与える定数を C_2 とする．

集合 $SUBG_{TAIL}$ について，$\hat{\Gamma}$ $(\{EXT_u[|EXT_u| - 1], EXT_u[|EXT_u|]\} \subseteq SUBG_u$ であることを考慮すれば，集合の定義は $O((|SUBG_u| + 1)^2) \leq O(|SUBG| \cdot (|SUBG_u| + 1))$ の手数で可能である．ここで上界 $C_3|SUBG| \cdot (|SUBG_u| + 1)$ を与える定数を C_3 とする．

もし $SUBG_{TAIL} = |SUBG_u| - 1$ であれば，EXPAND は $SUBG_u := \emptyset$ とする．この操作は $O(|SUBG_u| + 1)$ で行うことができるので，$C_4(|SUBG_u| + 1)$ を与える定数を C_4 とする．

(ii)$SUBG = \emptyset$ の場合

このとき，$SUBG_u = SUBG_{v_i} = \emptyset$ であるから，アルゴリズムはこれらの空集合性の判定を行った後，極大クリークを決定する．また，得られた極大クリークのサイズが既に得られた最大クリークのサイズより大であれば，最大クリークのサイズ $|Q\max|$ の更新を行う．

極大クリークのサイズについては，各深さにおいて節点が選出されるごとにクリークサイズの変数を $|ROOT|$ 分だけ加算，バックトラック時には減算すればよい．即ちクリークサイズは 1 語だけで格納可能である．

このように格納した2つのクリークサイズの比較は定数オーダーで可能であるから，クリークサイズの更新は適当な定数 C_5 で上界を与えることができる．

以上から，式 (1) における定数 C を

$$C = \max\{C_1, C_2, C_3, C_4, C_5\}$$

によって定義する．

また，EXPAND() の計算量 $T()$ は，その性質上対象とする節点集合のサイズに関して単調である．

主要定理証明に先立ち，先ず，EXPAND() による各子問題の時間計算量評価結果を以下に与える．

[補題 6]　節点集合が $SUBG$ で，最大次数節点 u の次数が $|SUBG_u| \geq 0$ なる任意のグラフにおいて，EXPAND($SUBG_u$), EXPAND($SUBG_{v_i}$)，および EXPAND($SUBG_{TAIL}$) の最大時間計算量は，$C' = 10^6 \cdot 2C$ とすると，それぞれ，

$$T(|SUBG_u|) \leq C' 2^{0.369894|SUBG_u|}(|SUBG_u|+1)^2,$$

$$T(|SUBG_{v_i}|) \leq C' 2^{0.369894|SUBG_{v_i}|}(|SUBG_{v_i}|+1)^2 \quad (1 \leq i \leq |EXT_u| - 2),$$

および，

$$T(|SUBG_{TAIL}|) \leq C' 2^{0.369894|SUBG_{TAIL}|} \cdot (|SUBG_{TAIL}|+1)^2.$$

(証明)　以下，最初に $T(|SUBG_u|)$ について，補題の関係の成立を示す．証明は最大次数 $|SUBG_u|$ に関する数学的帰納法による．

先ず，$|SUBG_u| = 0$ の場合，$SUBG_u = \emptyset, SUBG_{v_i} = \emptyset$ $(1 \leq i \leq |EXT_u|)$ である．ここで節点 u_1 を G の $SUBG_u$ による誘導部分グラフ中の最大次数節点とすれば，定数 C の定義により，

$$\begin{aligned} T(|SUBG_u|) &\leq C|SUBG_u| \cdot (|SUBG_{u_1}| + 1) \\ &< C' \cdot 1 \cdot (|SUBG_u| + 1) \cdot (|SUBG_{u_1}| + 1) \\ &= C' \cdot 2^{0.369894 \cdot 0} \cdot (|SUBG_u| + 1)^2. \end{aligned}$$

従って，題意は成立する．

この場合，$T(|SUBG_{v_i}|)$, $T(|SUBG_{TAIL}|)$ についても同様に題意は成立する．

次に，ある $|SUBG_u| \geq 0$ 以下の全ての $|SUBG_u|$ において補題中の全不等式が成立すると仮定する．この仮定のもとで，$SUBG$ 中の最大次数節点 u' の次数が $|SUBG_{u'}| = |SUBG_u| + 1$ となる $SUBG$ について要する計算量を考える．

この仮定のもとで，$SUBG_{u'}$ 中の最大次数節点 u_1 の次数は高々$|SUBG_u|$ である．そこで，節点 u_1 の隣接節点集合 $SUBG_{u_1}$ のサイズを $|SUBG_{u_1}| = |SUBG_u| - k$ $(0 \leq k \leq |SUBG_u|)$ と表す．このとき，式 (1) より，

$$\begin{aligned} T(|SUBG_{u'}|) &= T(|SUBG_u| + 1) \\ &\leq T(|SUBG_u| - k) \\ &\quad + \textstyle\sum_{i=1}^{|EXT_{u'}|-2} T(|SUBG_{v_i}|) \\ &\quad + T(|SUBG_{TAIL}|) \\ &\quad + C(|SUBG_{u'}| + 1) \cdot (|SUBG_{u_1}| + 1). \end{aligned}$$

節点 u_1 は $SUBG_{u'}$ 中で最大次数の節点であるから，$SUBG_{u'}$ における $v_1, v_2, \ldots, v_{|EXT_{u'}|-2}$ の次数は u_1 の次数以下である．従って，$|SUBG_{v_i}|$ $(1 \leq i \leq |EXT_{u'}| - 2)$ は $|SUBG_u| - k$ 以下である．ただし，$|SUBG_{TAIL}| = |SUBG_u| - 1$ の場合は $SUBG_u := \emptyset$ と代入を行う．このときは上記多項式項の定義により，$T(|SUB_u|) \leq C|SUBG_u|^2$ としてよい．

[1]$|EXT'_u| \geq 2$ である場合:

(a)$|SUBG_{TAIL}| \leq |SUBG_u| - 2$ の場合; $T()$ の単調性により，

$$\begin{aligned} &T(|SUBG_u| + 1) \\ &\leq ((|SUBG_u| + 1) - (|SUBG_u| - k) - 2) \cdot T(|SUBG_u| - k) \\ &\quad + T(|SUBG_{TAIL}|) \\ &\quad + C(|SUBG_u| + 1) \cdot (|SUBG_{u_1}| + 1). \end{aligned}$$

ここで，帰納法の仮定により，

$$\begin{aligned} &T(|SUBG_u| + 1) \\ &\leq (k-1) \cdot C' 2^{0.369894(|SUBG_u|-k)} \cdot ((|SUBG_u| - k) + 1)^2 \\ &\quad + T(|SUBG_{TAIL}|) \\ &\quad + C(|SUBG_u| + 1) \cdot (|SUBG_{u_1}| + 1) \\ &< C' 2^{0.369894(|SUBG_u|)} \left(\tfrac{k-1}{2^{0.369894k}}\right) \cdot (|SUBG_u| + 2)^2 \\ &\quad + T(|SUBG_{TAIL}|) + C(|SUBG_u|+2)^2. \end{aligned}$$

いま，$|SUBG_{TAIL}| = ((|SUBG_u| - k) - 2$. 従って，

$$\begin{aligned} &T(|SUBG_u| + 1) \\ &\leq C' 2^{0.369894(|SUBG_u|)} \left(\tfrac{k-1}{2^{0.369894k}}\right) \cdot (|SUBG_u|+2)^2 \end{aligned}$$

$+C'2^{0.369894(|SUBG_u|-k-2)}(((|SUBG_u| - k) - 2) + 1)^2 + 2C(|SUBG_u| + 2)^2$

$\leq C'2^{0.369894(|SUBG_u|)}\left(\frac{k-(1-\frac{1}{2^{2\cdot 0.369894}})}{2^{0.369894k}} + 10^{-6}\right)(|SUBG_u| + 2)^2$

ここで，文献 [4] と同様にして，$k \geq 0$ において $\frac{k-(1-\frac{1}{2^{2\cdot 0.369894}})}{2^{0.369894k}} < 1.29225$ であることを証明できる (付録，[補題 A1])．従って，

$T(|SUBG_u| + 1)$

$\leq C'2^{0.369894|SUBG_u|}(1.29225 + 10^{-6}) \cdot (|SUBG_u| + 2)^2$

$= C'2^{0.369894|SUBG_u|} \cdot 1.292251 \cdot (|SUBG_u| + 2)^2$

$< C'2^{0.369894|SUBG_u|} \cdot 2^{0.369894} \cdot (|SUBG_u| + 2)^2$

$= C'2^{0.369894(|SUBG_u|+1)}((|SUBG_u| + 1) + 1)^2.$

(b)$|SUBG_{TAIL}| = |SUBG_u| - 1$ の場合;

$SUBG_u = \emptyset$ と設定しているから

$T(|SUBG_u| + 1)$

$\leq ((|SUBG_u| + 1) - (|SUBG_u| - k) - 2 - 1) \cdot T(|SUBG_u| - k)$

$+C(|SUBG_u| + 1)^2$

$+T(|SUBG_{TAIL}|) + C(|SUBG_u| + 1)^2.$

従って，

$T(|SUBG_u| + 1)$

$\leq (k-2)T(|SUBG| - k)$

$+T(|SUBG_{TAIL}|) + 2C(|SUBG_u| + 1)^2.$

$|SUBG_{TAIL}| = |SUBG_u| - 1$ であるから，

$T(|SUBG_u| + 1)$

$\leq (k-2) \cdot C'2^{0.369894(|SUBG_u|-k)} \cdot ((|SUBG_u| - k) + 1)^2$

$+C'2^{0.369894(|SUBG_u|-k-1)}$

$+(C+1)(|SUBG_u| + 2)^2$

$= C'2^{0.369894(|SUBG_u|)}\left(\frac{k-(2-\frac{1}{2^{0.369894}})}{2^{0.369894k}} + 10^{-6}\right)(|SUBG_u| + 2)^2$

$= C'2^{0.369894(|SUBG_u|)}\left(\frac{k-(1+(1-\frac{1}{2^{0.369894}}))}{2^{0.369894k}} + 10^{-6}\right)(|SUBG_u| + 2)^2$

$< C'2^{0.369894(|SUBG_u|)}\left(\frac{k-(1-\frac{1}{2^{2\cdot 0.369894}})}{2^{0.369894k}} + 10^{-6}\right)(|SUBG_u| + 2)^2.$

よって上記と同様に成立する．

[2]$|EXT_{u'}| \leq 1$ である場合:

u' の子節点のうち探索されるのは高々u_1 のみである．従ってこの場合，

$T(|SUBG_u| + 1) \leq T(|SUBG_{u_1}|)$

$+C(|SUBG_u| + 1) \cdot |SUBG_{u_1}|$

$\leq C'2^{0.369894|SUBG_u|}(|SUBG_u| + 1)^2$

$+C(|SUBG_u| + 2)^2$

$\leq C'2^{0.369894|SUBG_u|}\left(1 + \frac{1}{10^6 \cdot 2^{0.369894|SUBG_u|}}\right) \cdot (|SUBG_u| + 2)^2$

$\leq C'2^{0.369894(|SUBG_u|+1)}((|SUBG_u| + 1) + 1)^2$

であり，やはり題意は成立する．

以上，帰納法による帰結により，任意の $|SUBG_u| \geq 0$ において，EXPAND($SUBG_u$) についての題意の関係が成立する．

EXPAND($SUBG_{v_i}$) および EXPAND($SUBG_{TAIL}$) についても，全く同様にして題意の成立を示すことができる．

以上より，帰納法帰結として，補題が証明された．□

これより，次が成立する．

[補題 7] 節点数 n，最大次数 $\Delta \geq 0$ なるグラフにおいて定数 C'' を $C'' = 1.2 \cdot 10^{11}$ とする．このとき EXPAND($SUBG$) の最大時間計算量 $T(n) = T(|V|)$ は次を満たす．

$T(|V|) = T(n) \leq C'C''2^{0.3827\Delta}n + Cn \cdot (\Delta + 1).$

(証明) 式 (1) より，

$T(n) \leq T(|SUBG_u|) + \sum_{i=1}^{|EXT_u|-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + Cn \cdot (\Delta + 1)$

である．ここで [補題 6] を用いれば

$T(n) \leq (n - \Delta - 1)C'2^{0.369894\Delta}(\Delta + 1)^2 + Cn \cdot (\Delta + 1)$

$\leq nC'2^{0.369894\Delta}(\Delta + 1)^2 + Cn \cdot (\Delta + 1).$

定数 C'' の設定により，任意の Δ において $(\Delta + 1)^2 < C''2^{0.000006\Delta}$ が成立する (付録，[補題 A2]) から，

$T(n) < nC'C''2^{0.369894\Delta} \cdot 2^{0.000006\Delta} + Cn \cdot (\Delta + 1)$

$= C'C''2^{0.3699\Delta}n + Cn \cdot (\Delta + 1)$ □

以上から，本論文の結論となる次の定理を得る．

[定理] 節点数 n のグラフにおいて，グラフの最大次数 Δ が

$\Delta \leq 2.704d\lg n$ ($d \geq 0$:定数)

を満すならば，最大クリーク問題は $O(n^{1+d})$ 時間で解決可能である．

(証明) 補題 7 において $\Delta \leq 2.704d\lg n$ とすれば

$T(n) \leq n \cdot C'C''2^{0.3699 \cdot 2.704 d \lg n} n + Cn \cdot (2.704 d \lg n + 1)$
$\leq nC'C''n^d + 2.704Cn \cdot \lg n^d + Cn$
$\leq C'C''n^{1+d} + 2.704Cn^{1+d} + Cn^{1+d}$
$= (C'C'' + 3.704C)n^{1+d}$. □

5 むすび

本稿では，文献 [7] の更なる改良結果として，節点数 n のグラフにおいて最大次数 $\Delta \leq 2.704 d \lg n$ ($d \geq 0$:定数) であるならば，最大クリーク問題は $O(n^{1+d})$ なる多項式時間で解決可能であることを示した．この結果は，最大クリーク抽出の基本アルゴリズム [4] 中に "部分問題の統合による探索領域削減" という手法を導入し，解析を更に詳細化した上でアルゴリズムを改良することにより達成している．ここでの結果は，アルゴリズム，解析の双方において，文献 [4] の単純さをほぼ保っている．アルゴリズムの単純さは，文献 [9] と同様，実用上の高速性にもつながるものである．

本稿の結果は，最大クリーク問題を容易に (多項式時間的に) 解ける範囲を従来よりも一層拡張できたことを示した．特に，グラフの次数に関する前提における定数 d に対する条件を，$d \geq 1$ から $d \geq 0$ へと拡張したことにより，グラフが疎で d が 1 よりも小さくなるに従い，所用計算時間オーダは節点数 n の 2 乗よりも小さくなる． この結果は，多くの応用例 ([2], [10], [11], 他) との関係においても重要である．

今後の課題は，アルゴリズムと解析双方の単純性を出来る限り保ちながら，最大クリーク問題の定量的時間計算量に関する結果を更に改良することである．

謝辞 文献 [7] の内容に関して貴重なご意見をいただいた，電通大・垂井淳 准教授をはじめ電子情報通信学会コンピュテーション研究会各位に深謝します．また，ご支援・協力をいただいた電通大・先進アルゴリズム研究ステーション高橋治久 教授，北大・原口誠 教授，他関係者に感謝します．なお，本研究は科研費基盤 (B), (C)，および総務省 SCOPE による研究助成を受けている．

参考文献

[1] M.R. Garey and D.S. Johnson, Computers and Intaractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, New York, NY, USA, 1979.

[2] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," in: D.-Z. Du and P. M. Pardalos (Eds.). Handbook of Combinatorial Optimization, Supplement vol.A, Kluwer Academic Publishers, pp.1–74, 1999.

[3] M. Gröstel, L. Lovász, and A. Schrijver, "Polynomial algorithm for perfect graphs," Annals of Discrete Math., vol.21, pp.325–356, 1984.

[4] 中西裕陽, 富田悦次, "最大クリーク問題の多項式時間的可解性の一結果," 信学論 (D), vol.J93-D, no.4, pp.417-425, Apr. 2010.

[5] 松野浩嗣, 田中都子, "定数次数のグラフの最大クリークを抽出するビット演算アルゴリズム," 情処学論, vol.37, pp,1869–1872, 1996.

[6] E. Tomita, A. Tanaka and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," Theoretical Computer Science, vol.363, pp.28-42, 2006.

[7] 中西裕陽, 富田悦次, "最大クリーク問題の多項式時間的可解性の改良結果," 信学技報, COMP2010-43, pp.29-36, Dec.. 2010 (信学論 (D) 採録決定).

[8] C. Bron and J. Kerbosch, "Algorithm 457, Finding all cliques of an undirected graph," Commun. ACM, vol.16, pp.575–577, 1973.

[9] 新道美喜男, 富田悦次, "最大クリークを抽出する単純なアルゴリズムとその最大時間計算量, " 信学論 (D), vol.J71–D, pp.472-481, March 1988.

[10] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi and M. Wakatsuki, "A simple and faster branch-and-bound algorithm for finding a maximum clique," WALCOM 2010, LNCS 5942, pp.191–203, 2010.

[11] E. Tomita, T. Akutsu, and T. Matsunaga, "Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics," in "Biomedical Engineering, Trends in Electronics, Communications and Software, " Anthony N. Laskovski (Ed.), ISBN: 978-953-307-475-7, InTech, pp.625-640, 2011. Available from: http://www.intechopen.com/articles/show/title/efficient-algorithms-for-finding-maximum-and-maximal-cliques-effective-tools-for-bioinformatics